(紹介) REDUCE-2プログラムについて

京大・教養 富 田 博 之

(1979年1月16日受理)

§1. はじめに

先月号の編集後記で電卓や計算機のことをダシに使った罪ほろぼしに，最後に出てきたREDUCE-2プログラムについての紹介を試みることにした。冬休みを利用して実行してみた二，三の例題に即して，FORTRANに比べどういうことが可能か，を示すにとどめたいが，まだ十分に使いこなせていないし，前身であるLISPにも通じていないため，いささか偏った紹介になることは否定できないし，不正確な説明もあることはお許し願いたい。完全なマニュアルはまだ手に入らないが，現存する解説としては文献1，2およびその引用文献を参照されたい。なお例題の計算には京大大型計算機センターのFACOM M-190を使用した。

FORTRANが数値計算処理言語で，いわば算数であるのに対し，REDUCE-2は数式処理用言語で，次のような代数計算を処理できる。

① 数式の展開，簡約化，変形

② 変数や数式の代入による変換

③ 数式微分演算

④ 行列計算(四則演算，転置，跡，逆行列，行列式)

もちろん，因数分解一般や不定積分は不可能である。その他にLISPによる処理やおそらく開発者による先祖の記憶が本能的に残っているのか，高エネルギー物理学におけるスピン1/2，1を含む代数計算が可能とされている。出力形式を制御するいろんなフラグが用意されており最も目的にかなった書式法を選ぶことができるが，以下の例では原則としてすべてのフラグは自然のまま(指定しない時の形)にしてある。各種のメッセージも(エラーメッセージはない)カットしてある。

§2. 代数計算(角運動量演算子の積の跡計算)

富田博之

磁性体の熱力学的量の高温展開等に使われる

$$\mathrm{Tr}\,(J^{-}J^{+}J^{z}J^{-}J^{z}J^{+})$$

のような計算で，すでに文献3，4に作表されている。おそらくFORTRAN で計算されたものと思われるが，ここではREDUCE -2で代数計算を試みる。まずプログラムに入る前にプログラム中の変数を用いて手順を示しておこう。

(i) まず

$$Z(n) = \mathrm{Tr}\,(J^{z})^{n}/(2J+1)$$

を求める。ただし角運動量の大きさはJとし以後**文字のまま**で扱われる。

$$(M+1)^{n+1} - M^{n+1} = (n+1)M^{n} + 1 + \sum_{k=1}^{n-1} C(n+1,\ k)M^{k}$$

を用いれば

$$Z(n) = \sum_{M=-J}^{J} M^{n}/(2J+1)$$

に対し漸化式

$$Z(n) = \{[(J+1)^{n+1} - (-J)^{n+1}]/(2J+1) - 1$$

$$- \sum_{k=1}^{n-1} C(n+1,\ k)\,Z(k)\}/(n+1),$$

$$Z(0) = 1,\quad Z(1) = 0$$

を得る。ただし，

$$C(n+1,\ k) = (n+1)!/k!\,(n+1-k)!$$

$$= \prod_{i=1}^{k} (n+1-k+i)/i$$

である。これを文献3に習って

$$X = J(J+1)$$

の多項式で表わしておく。

(ii) J^z, $J^{\pm} = J^x \pm i J^y$ の行列要素

$$\langle M|\, J^z \,|M'\rangle = M \qquad \text{for} \quad M' = M$$
$$\langle M|\, J^- \,|M'\rangle = \sqrt{(J-M)(J+M+1)} \qquad \text{for} \quad M' = M+1$$
$$\langle M|\, J^+ \,|M'\rangle = \sqrt{(J+M)(J-M+1)} \qquad \text{for} \quad M' = M-1$$

は，J^z, J^-, J^+ に整数 L＝0, 1, 2 を対応させ[3]，配列変数 MM(L)；

$$MM(0) = 0, \quad MM(1) = 1, \quad MM(2) = -1$$

および函数 Y(L, M)；

$$Y(0, M) = M$$
$$Y(1, M) = (X - M - M^2)^{1/2} \qquad (X = J(J+1))$$
$$Y(2, M) = (X + M - M^2)^{1/2}$$

を導入して

$$\langle M|\, L \,|M + MM(L)\rangle = Y(L, M)$$

と書ける。これより角運動量演算子の積

$$L(1)\, L(2) \cdots\cdots\cdots L(P)$$

の対角要素はこれをくり返し用いて

$$\langle M|\, L(1)\, L(2) \cdots\cdots\cdots L(P) \,|M\rangle$$
$$= \langle M + MI(P) \,|M\rangle \prod_{k=1}^{P} Y(L(k), M + MI(k-1))$$

ただし

$$MI(k) = \sum_{i=1}^{P} MM(i)$$

となる。MI(P) ≠ 0 の時，この対角成分はゼロであり，MI(P)＝0 の時，これはMについて高々P次の多項式となるのでこれを展開し M^n の係数を F(n) (Xの多項式) と

すれば,

$$\mathrm{Tr}\,(L(1)\,L(2)\cdots\cdots L(P)) = (2J+1)\sum_{n=0}^{P} F(n)Z(n)$$

を得る。プログラム中では，P次の積に対しては0, 1, 2から成るP桁の整数JJ を対応させ，例えばP=6，JJ=120210 は，

$$T(120210,\,6) = \mathrm{Tr}\,(J^-J^+J^zJ^+J^-J^z)/(2J+1)$$

の対応を示す。以上の手順(i)(ii) のプログラムを表1, 2に示した。 会話型で行っているため，出力は出力命令のすぐ次に行われていることに注意されたい。また各行の左端の数字および・印は説明のために付け加えたものである。なお例は一応12次までを 想定してある。変数名はFORTRANに準じるが，いくつか予約語がある。

(プログラムの説明)……番号は表の左端の番号

1. 注釈文であり実行されない。最後の；は文のくぎり符である。くぎり符には；または$(または¥)が使われ，評価の行われる文で結果を出力させたい時に；を用いる以外にはどちらでもよい。くぎり符さえあれば，文は一行にいくら書いてもよいし，何行にわたってもよい。
2. LET文は，以後左辺の表式が現われた時，右辺の表式でおきかえる。 ここでは変数JがX=J(J+1)におきかえられるが，左辺の表式としては今のところ*(積)と**(ベキ)を含むものしか許されておらず，

   LET　J*(J+1)=X;

   は不可なのでこのようにした。なお,

   LET　Y=Y+A;

   のような代入文も，無限に代入がくり返されるので許されない。
3. 配列宣言とその中身の定義をまとめてある。配列の引数はFORTRANとちがい0まで許される。FOR文はFORTRANのDOループに相当し，ひとつの文にまとめられるので便利である。N:=1:NN は

表1. Tr(J^z)^n を求めるプログラムと結果(左端の・印は出力行を示す)

```
1 COMMENT *** Z(N)=Tr(JZ**N)/(2J+1) ***;
2 LET J*J=X-J;
3 ARRAY Z(12); Z(0):=1$ Z(1):=0$ FOR N:=2:12 DO
          Z(N):=(((J+1)**(N+1)-(-J)**(N+1))/
             (2*J+1)-1-(FOR K:=1:N-1 SUM Z(K)*
        (FOR I:=1:K PRODUCT (N+1-K+I)/I)))/(N+1)$

4 FOR N:=0:12 DO WRITE "Z(", N, ")=", Z(N);

5 • Z(0)=1

  • Z(1)=0

  • Z(2)=X/3

  • Z(3)=0

  • Z(4)=(X*(3*X - 1))/15

  • Z(5)=0

                   2
  • Z(6)=(X*(3*X   - 3*X + 1))/21

  • Z(7)=0

                   3          2
  • Z(8)=(X*(5*X   - 10*X   + 9*X - 3))/45

  • Z(9)=0

                    4           3          2
  • Z(10)=(X*(3*X   - 10*X   + 17*X   - 15*X + 5))/33

  • Z(11)=0

                        5          4            3            2
  • Z(12)=(X*(105*X   - 525*X   + 1435*X   - 2360*X   +

               2073*X - 691))/1365
```

表 2. 一般積項を求めるプログラム (表 1 より継続)

```
   COMMENT *** Tr(J(1)J(2)...J(P))/(2J+1) ***;
   ARRAY MM(2); MM(0):=0$ MM(1):=1$ MM(2):=-1$
6  OPERATOR Y; FOR ALL M LET Y(0,M)=M,
            Y(1,M)=(X-M-M**2)**(1/2),
            Y(2,M)=(X+M-M**2)**(1/2)$
7  ARRAY L(12), MI(12), F(12);
8  ALGEBRAIC PROCEDURE T(JJ,P);
9   BEGIN FOR K:=1:P DO BEGIN N:=1$
10     A: JK:=N*10**(P-K)$ N:=N+1$
11        IF JJ >= JK THEN GO TO A;
12     L(K):=N-2$ JJ:=JJ-(N-2)*10**(P-K) END$
       MI(0):=0$ FOR K:=1:P DO
       MI(K):=MI(K-1)+MM(L(K))$
13     IF MI(P) NEQ 0 THEN RETURN;
       YY:=FOR K:=1:P PRODUCT Y(L(K),M+MI(K-1))$
14     COEFF(YY,M,F)$
       TT:=FOR N:=0:P SUM F(N)*Z(N)$
15     RETURN TT;
16  END$

17 T(00,2);
18 X/3

   T(0000,4);
   (X*(3*X - 1))/15

   T(12121212,8);
               3       2
   (16*X*(8*X  + 8*X  + 3*X + 3))/315

   T(1212121,7);
   0

   T(12120000,8);
             3        2
   (2*X*(4*X  + 22*X  - 6*X - 3))/315

   T(1201201200,10);
             4        3          2
   (2*X*(8*X  + 54*X  + 104*X  - 117*X + 28))/1155
```

N: = 1 STEP 1 UNTIL NN

の省略形(STEPが1の時のみ)であり，この後には

$$\left\{\begin{array}{l}\text{DO+実行文(代入，出力等)}\\ \text{SUM}\\ \text{PRODUCT}\end{array}\right.$$

が続く。従って例えば

FOR K: = 1 : N SUM…… = $\sum_{K=1}^{N}$ ……

と考えればよい。: =は代入を実行する演算子で例えば

Y: = (X+1)**2 $ X: = 3 $ Y;

とすればYの値として16が出力される。 FORTRANでは函数定義の場合を除いて文字Xが現われるより前にXが表すべき数値が定義されていなければならないのに対し，このような使い方ができる点が異る。なお，この文で一度評価が実行されるとその結果は自動的に保管されており，以後にZ(N)を引用する時には評価はくり返されない。

4. FOR文を用いた出力命令。文字列を出力したい時は″で囲む。
5. 以下ただちに出力が行われる。WRITE文を実行する毎に行が変わる。 FORTRANと異りベキは出力の際には普通の形で書かれる。(FORTRAN 形の出力制御フラグもある。)
6. 演算子(ここでは函数)の宣言とその定義。 FOR ALL MはMが変数であることを示している。これがない時にはLET文の左辺にある**表式そのものの形**が現われた時にしか代入を実行せず，例えばY(0, M+2)に対しては= M+2 の代入は行われない。
7. 配列宣言。これは用いられる直前に宣言すればいいが，ここで行っておかないと8以下の手続き文を使用する度に「既に宣言されている」としつこく警告される。また宣言を省略すると，引数つきのため「演算子かどうか」を尋ねてくる。(Fについて

は 14 参照)

8. 後でＪＪ，Ｐの具体的な数値を用いてＴ(ＪＪ，Ｐ)を入力した時，その内容を評価するための代数手続の宣言で，内容は続く 9 以下の文 (今の場合複数から成る複合文) で定義される。

9. BEGIN は複合文の開始を示し，最初の BEGIN から 16 の END までの複合文で代数手続きが定義される。二番目の BEGIN は，FOR 文の実行文がやはり複合文で，12 の END までをひとまとめにＫ= 1, 2, ……， Ｐと実行するためのもので，全体で二重の複合文になっている。

10. A: は文のラベルで， GO TO で行先を指定したい時につけられる。

11. ＩＦは FORTRAN と同じく分岐を形成し，次の判断 (この例ではＪＪ ≧ ＪＫ)が正の時， THEN 以下を実行，誤の時，ここでは ELSE 文が省略されているので次の文へ行く。

12. 9 ～ 12 で整数ＪＪ (例えば 0012) から配列 L(P) (P= 4, L(1) = 0，L(2) = 0, L(3) = 1, L(4) = 2) を作る。 FORTRAN のＩＦＩＸのような演算が組みこまれていないのでやっかいである*)。

13. RETURN は手続複合文から抜け出る命令で引数なしの時， Ｔの評価値ゼロを持帰る。

14. COEFF(YY，M, F) は YY を M について展開した M^nの係数を用意された (7 で宣言) 配列要素 F(n) に代入する演算子である。配列宣言が行われていなければ，新たに Fn という形の変数が造成される。

15. 引数つき RETURN は引数 TT の値を T の評価値として持帰る。

16. 手続き複合文の終りを示す。

17. 以下対話形式で入出力が行われる。；は出力命令を伴ったくぎり符である。

以上の計算では TSS 端末使用で約 20 秒かかっており， FORTRAN に比べてケタちがいに時間がかかるようであり，文献 3 のような作表は大変である。その代りにプログラムは簡単であり，以下の例のように FORTRAN ではできない相談もできるようである。

---

*) 一時的にＬＩＳＰに移れば簡単になる。

§ 3. 行列計算（角運動量 J = 1 演算子）

表3のように行列宣言を行うことにより，行列計算が可能である。次元指定の行列宣言により全要素は0に設定されるので，0でない要素だけ再定義してある。TP は転置演算子，DET は行列式演算子である。なお跡演算 TRACE はこの時点では使用不可で

表3. 行列計算（角運動量 J ＝ 1）

```
COMMENT *** Matrix Formulation for J ***
                (J0=Jz, J1=J-, J2=J+);
J:=1$ JJ:=2*J+1$
MATRIX A(JJ,JJ), J0(JJ,JJ), J1(JJ,JJ), J2(JJ,JJ);
  COMMENT  N=J-M+1 ;
  FOR N:=1:JJ DO J0(N,N):=J-N+1$ FOR N:=2:JJ
  DO J1(N,N-1):=((N-1)*(2*J-N+2))**(1/2)$
  J2:=TP J1$
PROCEDURE TR(A); FOR N:=1:JJ SUM A(N,N);
ON NERO; J0*J1-J1*J0;
                        (1/2)
MAT(2,1) :=  - 2


                        (1/2)
MAT(3,2) :=  - 2

J0*J2-J2*J0;
                   (1/2)
MAT(1,2) := 2


                   (1/2)
MAT(2,3) := 2

J2*J1-J1*J2;
MAT(1,1) := 2

MAT(3,3) := (-2)

OFF NERO; DET((J1*J2+J2*J1)/2+J0**2);
8

A:=(J1*J2)**4$ TR(A)/JJ;
32/3

A:=(J1*J2*J0)**3*J0$ TR(A)/JJ;
8/3
```

あったので，オペレーターTRとして手続き規定をしてある。NEROは値が0の要素を出力させないためのフラグである。（自然にはOFF状態）行列要素はもちろん，文字のままでもよく，逆行列演算（逆べキ算）も可能なので，連立一次方程式の解の一般形など簡単に出力できる。

§4. 微分演算（Legendre函数，Taylor展開）

変数Xを含んだ表式FのXに関する微分演算として

DF(F, X)；

がくみこまれている。M回微分 $d^M F / dX^M$ なら

DF(F, X, M)；

である。Fが代数式またはあらかじめ組みこまれている函数 (sin, cos, log) ならこれでよいが，自分で持ち込んだ函数の時はLET文で微分演算を定義する。表4，5に使い方を例示してある。表4は漸化公式を用いたLegendre多項式，陪函数の生成である。表5は，任意の函数FをXについてX＝AのまわりにTaylor展開しDX＝X－Aについて M次まで出力させる代数手続であり，二重の複合文が現われている。SUB(X＝A, F) は表式Fの中の変数XにAを代入する命令（およびその評価値）である。FACTOR DX；はDXについてくくり出す命令で，ここの例では不用である。またRATは部分分数形に出力させるフラグでこれがOFF（自然形）の時は，表2のような共通分母出力形になる。表5の三番目の例の函数TH(X) (tanh (x)) は，組み込まれていないので，微分公式をFOR ALL X LET文で，またTH(A)の値をLET文で定義しておかねばならない。

なお，多項式の場合には，例えばXの7次以上を捨てたければ

LET X**7＝0；

と入れればよい。二変数以上で，オーダーが各々異る時は例えば次のようにする。

WEIGHT X＝1, Y＝2；
WTLEVEL 3；

表 4. 微分演算（Legendre 多項式，陪函数）

```
COMMENT *** Legendre Functions ***;
OPERATOR P; FOR L:=0:3 DO BEGIN
IF L NEQ 0 THEN GOTO A ELSE P(L,0,X):=1$
WRITE "P(0,0,X) := 1"; GOTO B;
A: WRITE P(L,0,X):=X*P(L-1,0,X)
                -(1-X**2)*DF(P(L-1,0,X),X)/L;
  FOR M:=1:L DO WRITE P(L,M,X):=
                (1-X**2)**(M/2)*DF(P(L,0,X),X,M);
B: END;

P(0,0,X) := 1

P(1,0,X) := X

                  2      (1/2)
P(1,1,X) := ( - X  + 1)

                 2
P(2,0,X) := (3*X  - 1)/2

                    2      (1/2)
P(2,1,X) := 3*( - X  + 1)      *X

                    2
P(2,2,X) := 3*( - X  + 1)

                    2
P(3,0,X) := (X*(5*X  - 3))/2

                      2      (1/2)      2
P(3,1,X) := (3*( - X  + 1)      *(5*X  - 1))/2

                        2
P(3,2,X) := 15*X*( - X  + 1)

                      2      (1/2)       2
P(3,3,X) := 15*( - X  + 1)      *( - X  + 1)
```

表5.　Taylor 展開

```
COMMENT *** Taylor Expansion ***;
ALGEBRAIC PROCEDURE TAYLOR(F,X,A,M);
BEGIN FACTOR DX; FF:=0$ XX:=1$
   FOR N:=0:M DO BEGIN FF:=FF+SUB(X=A,F)*XX$
   F:=DF(F,X)$ XX:=XX*DX/(N+1) END$
RETURN FF END;
ON RAT;


Y:=TAYLOR(COS(X),X,0,6);
              6             4                2
•Y := ( - DX )/720 + DX /24 + ( - DX )/2 + 1

F:=(1-X**2)**(1/2)$ Y:=TAYLOR(F,X,0,6);
              6               4                2
•Y := ( - DX )/16 + ( - DX )/8 + ( - DX )/2 + 1

OPERATOR TH; FOR ALL X LET DF(TH(X),X)=1-TH(X)**2;
LET TH(0)=0; TAYLOR(TH(X),X,0,7);
           7              5                3
•( - 17*DX )/315 + (2*DX )/15 + ( - DX )/3 + DX
```

この時，1, X, X^2, Y, X^3, XY 以外は捨てられる。

組みこまれている函数については表6に示した。sin, cos, log については導函数もくみこまれている。なお指数函数は FORTRAN の EXP(X) ではなく，底 E が予約変数として組みこまれており，e^X＝E**X である。予約変数としては虚数単位 I すなわち I**2＝−1 も組みこまれている。

表6. 組みこまれている函数と指数函数

```
 COMMENT * SIN(X) *; Y:=SIN(0); Y1:=DF(SIN(X),X);
•Y := 0

•Y1 := COS(X)

 COMMENT * COS(X) *; Y:=COS(0); Y1:=DF(COS(X),X);
•Y := 1

•Y1 :=  - SIN(X)

 COMMENT * LOG(X) *; Y:=LOG(1); Y1:=DF(LOG(X),X);
•Y := 0

•Y1 := 1/X

 COMMENT * EXP(X) *; Y:=E**X$ Y1:=DF(Y,X);
           X
•Y1 := E
```

§5. おわりに

まだいくつか便利な操作が残されているが，この辺で時間ぎれとしたい。感想としては，今のところ制限がかなりあるのと，時間をくうことが難点のようである。例えば量子力学で必要な順序づけされた非可換積演算など簡単な表現で表わせそうにない*)。また多項式以外の漸近評価も簡単な命令は組みこまれていない。ベクトル解析に必要なgradやrotのように評価値がベクトル（(1, 3) マトリクス）になるオペレータも簡単には作れない。これに対し，オペランドがベクトルの

```
MATRIX A( 1, 3 ) ; FOR ALL A LET
DIV A= DF(A(1, 1), X) + DF(A(1, 2), Y) + DF(A(1, 3), Z) ;
```

または

*) 行列の積演算は要素が設定されていないと実行されない。

MATRIX A(1, 3) ; PROCEDURE DIV A;

DF(A(1, 1), X) + DE(A(1, 2), Y) + DF(A(1, 3), Z) ;

は可であった。(既に表3でも使われている。)なお最後にBoson Operatorの Normal Orderingの計算例を表7に示した。

表7. Boson Operator Normal Ordering

```
COMMENT *** Boson Operator.***(C=creation, A=anni-
           hilation, U=unit, N=number operator);
OPERATOR A,C,U,N;
FOR ALL X,Y LET A C X=C A X + U X, U A X=A X,
    A U X=A X, U C X=C X, C U X=C X, U U X=U X,
    A(X+Y)=A X + A Y, C(X+Y)=C X + C Y,
    U(X+Y)=U X + U Y, A(-X)=-A X,
    C(-X)=-C X, U(-X)=-U X;
Z1:=A A C A C C X; Z2:=A A A C C C X;

Z1 := 14*C(A(X)) + 8*C(C(A(A(X)))) + C(C(C(A(A(A(X
      )))))) + 4*U(X)

Z2 := 18*C(A(X)) + 9*C(C(A(A(X)))) + C(C(C(A(A(A(X
      )))))) + 6*U(X)
```

(注:実はUは不要,またNも使われていない。)

参 考 文 献

1) 金田康正「東京大学大型計算機センターニュース」Vol. 9 No. 2 ~ 12 (1977)

2) 竹尾賢一,渡辺豊英「京都大学大型計算機センター広報」Vol. 11 No. 4 (1978)

3) W. Marshall and S. W. Lovesey, Theory of Thermal Neutron Scattering (Oxford, Clarendon Press, 1971)

4) E. Ambler, J. C. Eisenstein and J. F. Schooley, J. Math. Phys. 3 (1962), 118.